



JP6112818

Biblio

Page 1



## PLL PRESET METHOD AND PRESET TYPE PLL CIRCUIT

Patent Number: JP6112818  
Publication date: 1994-04-22  
Inventor(s): ASADA KAZUhide  
Applicant(s): ICOM INC  
Requested Patent: ☐ JP6112818  
Application Number: JP19920261706 19920930  
Priority Number(s):  
IPC Classification: H03L7/10 ; H03L7/187  
EC Classification:  
Equivalents:

### Abstract

**PURPOSE:** To improve the C/N, simplify the circuit and to make preset data accurate.  
**CONSTITUTION:** The preset type PLL circuit comprising a phase comparator 1, a variable frequency divider 8, a VCO 2, a loop filter 4, a ROM 6, a RAM 7 and a CPU 5 is provided with a D/A converter 3 and an unlock detection means 9. Simultaneously at application of power, data corresponding to a preset frequency are read out and fed to a variable frequency divider 8, a VCO 2 is oscillated at a minimum frequency, then control data are changed by a specified value each and fed to the VCO 2 via D/A converter 3 and an unlock state is monitored by an unlock detection means. Then the data in unlock state are extracted and corrected by a data correction means and written in the RAM 7. The stored preset data are read out of the RAM 7 and inputted to the D/A converter 3 to control the VCO 2.

Data supplied from the esp@cenet database - I2

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-112818

(43)公開日 平成6年(1994)4月22日

(51)Int.Cl.<sup>5</sup>

識別記号

庁内整理番号

F I

### 技術表示箇所

H 0 3 L 7/10  
7/187

9182-5 J

H 0 3 L 7/ 10

D

9182-5 J

7/ 18

D

審査請求 未請求 請求項の数 3 (全 11 頁)

(21)出願番号 特願平4-261706

(22)出願日 平成4年(1992)9月30日

(71)出願人 000100746

アイコム株式会社

大阪府大阪市平野区加美鞍作1丁目6番19号

(72)発明者 浅田 和秀

大阪市平野区加美鞍作1丁目6番19号 ア  
イコム株式会社内

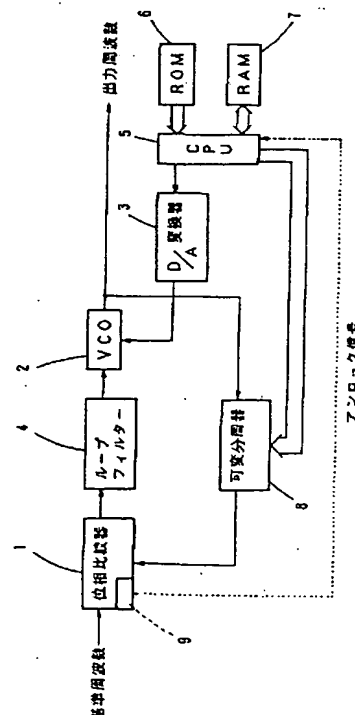
(74)代理人 弁理士 杉本 勝徳 (外1名)

(54)【発明の名称】 PLLプリセット方法およびプリセット型PLL回路

(57) 【要約】 (修正有)

【目的】 キャリア／ノイズ比の改善と回路の簡略化と  
プリセットデータの正確化をはかる。

【構成】 位相比較器 1 と可変分周器 8 と VCO 2 とループフィルタ 4 と ROM 6 と RAM 7 と CPU 5 からなるプリセット型 PLL 回路に、D/A 変換器 3 とアンロック検出手段 9 を設ける。電源投入と同時に、プリセット周波数に対応するデータを読み出し可変分周器 8 に加え、VCO 2 を最低周波数で発振させた後その制御データを規定値ずつ変化させて D/A 変換器 3 を介して VCO 2 に与え、アンロック状態をアンロック検出手段で監視する。そしてアンロックした時のデータを取り出し、データ補正手段で補正し、RAM 7 に書き込む。このようにして蓄積したプリセットデータを以後は RAM 7 から読みだし、D/A 変換器 3 に入力して VCO 2 を制御する。



1

## 【特許請求の範囲】

【請求項1】位相比較器と、可変分周器と、各分周比に対応するデータを記憶する記憶手段と、前記データによってプリセットされる電圧制御型発振器と、位相比較器がアンロック状態のときアンロック信号を出力するアンロック検出手段と、を備えたプリセット型PLL回路のPLLプリセット方法において、前記アンロック信号を監視しながら電圧制御型発振器への入力データを変更して、ロックレンジの上限と下限を検出し、前記ロックレンジの上限と下限に対応する電圧制御型発振器への入力データを出力し、出力された前記データを補正し、それによって得たデータを上記記憶手段に書き込むことを特徴とするPLLプリセット方法。

【請求項2】位相比較器と、可変分周器と、各分周比に対応するデータを記憶する記憶手段と、前記データによってプリセットされる電圧制御型発振器と、を備えたプリセット型PLL回路において、位相比較器がアンロック状態のとき、アンロック信号を出力するアンロック検出手段と、電圧制御型発振器への入力データを変更するデータ変更手段と、前記アンロック検出手段のアンロック信号が出力されなくなった時点と再び出力された時点において、電圧制御型発振器に与えられたデータを出力するデータ出力手段と、該データ出力手段から出力された前記データを補正し、それによって得たデータを上記記憶手段に書き込むデータ補正手段と、を備えていることを特徴とするプリセット型PLL回路。

【請求項3】位相比較器と、可変分周器と、各分周比に対応するデータを記憶する記憶手段と、前記データによってプリセットされる電圧制御型発振器と、を備えたプリセット型PLL回路において、位相比較器がアンロック状態のとき、アンロック信号を出力するアンロック検出手段と、電圧制御型発振器への入力データを変更するデータ変更手段と、前記アンロック検出手段のアンロック信号の立ち上がりの時点において、電圧制御型発振器に与えられたデータを出力するデータ出力手段と、該データ出力手段から出力された前記データを補正し、それによって得たデータを上記記憶手段に書き込むデータ補正手段と、を備えていることを特徴とするプリセット型PLL回路。

## 【発明の詳細な説明】

【0001】

2

【産業上の利用分野】本発明は通信機等に用いるPLL回路に関し、特にPLLプリセット方法およびプリセット型PLL回路に関するものである。

【0002】

【従来の技術】図7に示すようにPLL回路は、可変分周器18の分周比（以下Nデータという。）ごとに位相比較器11で位相を比較すると、位相差に応じた制御電圧が位相比較器11から出て、それをループフィルタ14で平滑した後、その電圧で電圧制御型発振器（以下VCOという。）12を制御し、周波数がロックするようにする。

【0003】従来から、PLL回路の発振周波数の高速切り換えの方法として、マイクロプロセッサ（以下CPUという。）15が可変分周器18を制御するとともに、記憶手段としてのランダムアクセスメモリー（以下RAMという。）17からのプリセットデータによって、デジタル／アナログ変換器（以下D/A変換器という。）13を介して直接、VCO12を制御するという動作を、プリセット周波数ごとにプリセットデータを次々に切り換えていくものがあつた。

【0004】この方式のPLL回路は、たとえば製造出荷時に、位相比較器がロックした時のVCO12の入力制御電圧を、アナログ／デジタル変換器（以下A/D変換器という。）10で読み取ってデジタル信号に直し、CPU15で機器の発熱の影響によるVCO12の発振時の誤差を考慮して、少し低めに補正した後、RAM17に記憶させ、それを必要に応じて取り出して、プリセットデータとして活用するものであつた。

【0005】なお、以上のCPU15による制御は、リードオンリーメモリー（以下ROMという。）16の制御プログラム、データによって行われる。

【0006】

【発明が解決しようとする課題】しかし、上述の従来の方法では、ロック時に高インピーダンスになっているVCOの入力制御電圧をA/D変換し、そのデータを取り出すために配線を引き回すので、外乱を受けやすく、却ってPLL回路全体としてのキャリア／ノイズ比（以下C/N比という。）の劣化の原因となる上に、A/D変換器やそれに付随するバッファアンプも必要で、回路が複雑になっていた。

【0007】また、機器の発熱の影響によるVCOの発振時の誤差を考慮して、プリセットデータの値を少し低めに補正した後、記憶し、それを使用するので、プリセットデータが不正確になりがちであつた。

【0008】本発明は、上記の問題点にかんがみて提案されたもので、C/N比の改善と回路の簡略化を図るとともに、より正確なプリセットデータを得ることを目的としている。

【0009】

50 【課題を解決するための手段】上記課題を解決するため

に、本発明のPLLプリセット方法においては、位相比較器と、可変分周器と、各分周比に対応するデータを記憶する記憶手段と、前記データによってプリセットされる電圧制御型発振器と、位相比較器がアンロック状態のときアンロック信号を出力するアンロック検出手段とを備えたプリセット型PLL回路のPLLプリセット方法において、前記アンロック信号を監視しながら電圧制御型発振器への入力データを変更して、ロックレンジの上限と下限を検出し、前記ロックレンジの上限と下限に対応する電圧制御型発振器への入力データを出力し、出力された前記データを補正し、それによって得たデータを上記記憶手段に書き込むという方法を用いた。

【0010】また、本発明のプリセット型PLL回路においては、位相比較器と、可変分周器と、各分周比に対応するデータを記憶する記憶手段と、前記データによってプリセットされる電圧制御型発振器とを備えたプリセット型PLL回路において、位相比較器がアンロック状態のとき、アンロック信号を出力するアンロック検出手段と、電圧制御型発振器への入力データを変更するデータ変更手段と、前記アンロック検出手段のアンロック信号が出力されなくなった時点と再び出力された時点において、電圧制御型発振器に与えられたデータを出力するデータ出力手段と、該データ出力手段から出力された前記データを補正し、それによって得たデータを上記記憶手段に書き込む補正手段とを備えた。

【0011】更に、位相比較器と、可変分周器と、各分周比に対応するデータを記憶する記憶手段と、前記データによってプリセットされる電圧制御型発振器とを備えたプリセット型PLL回路において、位相比較器がアンロック状態のとき、アンロック信号を出力するアンロック検出手段と、電圧制御型発振器への入力データを変更するデータ変更手段と、前記アンロック検出手段のアンロック信号の立ち上がりの時点において、電圧制御型発振器に与えられたデータを出力するデータ出力手段と、該データ出力手段から出力された前記データを補正し、それによって得たデータを上記記憶手段に書き込むデータ補正手段とを備えた。

【0012】

【作用】本発明のPLLプリセット方法およびプリセット型PLL回路においては、先ず所望周波数に対応するNデータが可変分周器に入力され、VCOにはそのVCOの最も低い発振周波数か若しくは高い発振周波数に対応するデータが入力される。その結果、VCOは最も低い周波数か若しくは高い周波数で発振し、その発振周波数は可変分周器で分周された後、位相比較器で基準周波数と位相が比較される。

【0013】この状態では、構成する素子のばらつき等のために位相比較器のロックがかかるロックレンジは不明であるが、入力されたデータは予想されるロックレンジから離れているのでロックせず、アンロック検出手段

によってアンロック信号が出力される。

【0014】次にVCOへの入力データをデータ変更手段によって変更していくと、やがて位相比較器のロックレンジに入りアンロック信号が出力されなくなる。この時点でのVCOへの入力データがデータ出力手段によって出力され、記憶手段に一時記憶される。

【0015】続いてそこからVCOへの入力データを変更していくと、位相比較器のロックレンジを外れ、再びアンロック信号が出力される。この時点でのVCOへの入力データがデータ出力手段によって出力され、記憶手段に一時記憶される。次に、一時記憶されたデータを取り出し、データ補正手段によって補正して、新しいデータを得、それを記憶手段に記憶させておく。それが次回からのプリセットデータとして使われる。

【0016】また、アンロック信号を監視しながら、データ変更手段によってVCOへの入力データを大きく変化させていくとやがてロックレンジに入り、アンロック信号の出力がなくなる。アンロック信号の出力がなくなった時点からは、VCOへの入力データをデータ変更手段によって、増加方向と減少方向に変更させていくと、ロックレンジの上端と下端の2箇所アンロック信号が現れる。その時のVCOの入力データがデータ出力手段によってそれぞれ出力され、記憶手段に一時記憶される。

【0017】そして、一時記憶されたデータを取り出し、データ補正手段によって補正して新しいデータを得る。

【0018】

【実施例】以下に本発明のPLLプリセット方法およびプリセット型PLL回路を、プリセット型PLL回路を使用した装置と図面に基づいて、詳細に説明する。

【0019】（実施例1）図1は本実施例のプリセット型PLL回路のブロック図、図2は図1のROMに内蔵された制御プログラムのフローチャート図、図3は図1のCPUからD/A変換器に出力されるデータとアンロック信号の出力のタイムチャート図である。ここで、CPUによって制御されるデータ変更手段とデータ出力手段とデータ補正手段のデータはデジタル信号であるが、図3の（B）のデータ値はそれをアナログ値になおして表している。

【0020】図中、符号1は位相比較器、2はVCO、3はD/A変換器、4はループフィルター、5はCPU、6はデータ変更手段とデータ出力手段とデータ補正手段のプログラムを含む制御プログラムとデータを内蔵したROM、7はRAM、8は可変分周器、9は位相比較器1に内蔵されたアンロック検出手段である。

【0021】図2のフローチャート図を中心に、図1と図3を参照して説明する。ステップS1において、プリセット周波数 $f_1$ をプリセットするために、ROM6からプリセット周波数 $f_1$ に対応するNデータ $N_1$ が読み

出され、可変分周器 8 に入力される。図 2 において、プリセット周波数  $f$  に対応する  $N$  データをデータ  $N$  とし、プリセット周波数  $f$  を決定するための変数データをデータ  $D(N)$  として、ここでは 1 回目のプリセット周波数  $f_1$  に対応するデータをデータ  $N_1$ 、データ  $D(N_1)$  としている。

【0022】なお、データ  $D(0)$  はプリセットデータ初期値で、一定の数値である。

【0023】次に、ステップ  $S_2$  において、図 3 (B) のタイミング  $T_1$  で、 $VCO_2$  固有の発振最低周波数に対応するプリセットデータ初期値  $D(0)$  が、データ  $D(N_1)$  を代入される。続いてステップ  $S_3$  で、データ  $D(N_1)$  に規定値  $d$  を足した後、ステップ  $S_4$  で、それを  $D/A$  変換器 3 に入力する。すると、 $D/A$  変換器 3 からはそれに応じた出力電圧が出て  $VCO_2$  に印加され、 $VCO_2$  は出力電圧に応じて発振する。

【0024】ここで、 $VCO_2$  は入力電圧が上がると、出力周波数も高くなるように構成されている。

【0025】 $VCO_2$  の出力周波数が可変分周器 8 に入ると、それが  $N$  データ  $N_1$  に応じて分周され、その周波数と基準周波数の位相を位相比較器 1 で比較する。データ  $D(N_1)$  は周波数  $f_1$  のロックレンジより十分に低い値なので、この時点で、位相比較器 1 に内蔵されたアンロック検出手段 9 は、アンロック状態を検出してアンロック信号を出力する。これが、ステップ  $S_5$  の段階である。

【0026】アンロック信号が出力された場合は、ステップ  $S_3$  に示すように、CPU 5 の指令によってデータ変更手段がデータ  $D(N_1)$  を規定値  $d$  だけ上げ、それを更新したデータ  $D(N_1)$  として、ステップ  $S_4$  で再び  $D/A$  変換器 3 に入力する。 $D/A$  変換器 3 の出力電圧が上がりそれが  $VCO_2$  に印加されると、 $VCO_2$  の発振周波数は上がる。その発振周波数は、上述の経路で位相比較器 1 で再びロック状態をチェックされる。

【0027】以上の動作を、アンロック信号が出力されなくなるまで繰り返す。このデータの加算の様子は図 3 の (B) に表されている。

【0028】ロックレンジに入ると、アンロック検出手段 9 は位相比較器 1 がロック状態になったことを検出する。そうならば図 3 (B) のタイミング  $T_2$  の時点で、データ出力手段は、CPU 5 内のデータ  $D(N_1')$  を読み取り、出力する。これをロックレンジの下端周波数のデータ  $L(N_1)$  として、RAM 7 に一時記憶しておく。これがステップ  $S_6$  である。

【0029】ロックレンジの下端周波数のデータ  $L(N_1)$  を記憶した後もデータは、データ  $D(N_1')$  から規定値  $d$  ずつ増加され、 $D/A$  変換器 3 に入力される。この状態が次にアンロック信号が検出されるまで続く。これがステップ  $S_7$ 、 $S_8$ 、 $S_9$  である。ロック状態では位相比較器 1、 $VCO_2$ 、ループフィルター 4、可変分

周器 8 が周波数をプリセット周波数に近づける動作をしている。

【0030】前記の動作を何回か繰り返した後、タイミング  $T_3$  の時点でアンロック信号  $U_2$  が出力されたら、それを受けたデータ出力手段は、ステップ  $S_{10}$  で、その時点で CPU 5 内のデータ  $D(N_1')$  を読み取って、これをロックレンジの上端周波数のデータ  $H(N_1)$  として、RAM 7 に一時記憶しておく。

【0031】タイミング  $T_3$  でロックレンジの上端周波数のデータ  $H(N_1)$  の値を得ると同時に、データの変更を中止する。次に、ステップ  $S_{11}$  で、ロックレンジの下端周波数のデータ  $L(N_1)$  から規定値  $d$  を引いた値と上端周波数のデータ  $H(N_1)$  を平均して平均値  $M(N_1)$  を出し、それをプリセットデータとしてステップ  $S_{12}$  で、RAM 7 に書き込む。ここで、図 3 (B) において  $RL$  が実際のロックレンジで、上記の平均値  $M(N_1)$  はロックレンジの中心値と殆ど誤差がない。

【0032】以上で一つのプリセット周波数  $f_1$  に対応する  $N$  データ  $N_1$  とプリセットデータ  $M(N_1)$  の書き込みが終了する。

【0033】更に、プリセットすべき周波数が残っていれば、ステップ  $S_{13}$  によって最初のステップ  $S_1$  の段階に戻り、タイミング  $T_4$  で、次のプリセット周波数  $f_2$  に対応するデータ  $N_2$  を読み込んで可変分周器 8 に入力し、ほぼ同時に  $VCO_2$  固有の発振最低周波数に対応するプリセットデータ初期値  $D(0)$  が、ステップ  $S_2$  で  $D(N)$  に代入され、ステップ  $S_3$  で規定値  $d$  が加算されて  $D/A$  変換器 3 に入力される。以上の動作を繰り返す。

【0034】いうまでもないが周囲の環境が同じであれば、プリセットデータ初期値  $D(0)$  は常に一定である。以上の動作の CPU 5 による制御は、ROM 6 に書かれた制御プログラム、データによって実行される。

【0035】なお、最初に  $D/A$  変換器 3 に入力するデータとして、 $VCO_2$  固有の発振最低周波数に対応するプリセットデータ初期値  $D(0)$  の代わりに  $VCO_2$  固有の発振最高周波数に対応するプリセットデータ初期値を入れてもよく、データ変更手段はその場合はデータを増加ではなく、減少するものにする。上端データと下端データは入れ替わるが、それ以外のデータ出力手段、データ補正手段は同じである。

【0036】また、 $VCO_2$  を制御するためのプリセットデータは、実施例の場合はデジタル値で、それを  $D/A$  変換器 3 によってアナログ値に変えているが、プリセットデータをアナログ値として、それによって直接  $VCO_2$  を制御してもよい。

(実施例 2)

【0037】また、上記の実施例 1 と構成は同じで、作用が若干異なる PLL プリセット方法およびプリセット型 PLL 回路について、図 4 と図 5 と図 6 に基づいて、

詳細に説明する。なお、図4と図5は実施例2の一連のフローチャートである。実施例1と同様の動作をするステップについては若干省略した。

【0038】ステップS21において、プリセット周波数 $f_{11}$ をプリセットするために、ROM6からプリセット周波数 $f_{11}$ に対応するNデータN11が読み出され、可変分周器8に入力される。ここで実施例1と同様に、プリセット周波数 $f$ に対応するNデータをデータNとし、プリセットデータを決定するための変数データをデータS(N)、データD(N)として、ここでは1回目のプリセット周波数 $f_{11}$ に対応するデータをデータN11、データS(N11)、データD(N11)としている。なお、データS(0)はプリセットデータ初期値で、一定の数値である。

【0039】次にステップS22でVCO2固有の発振最低周波数に対応するプリセットデータ初期値S(0)がデータS(N11)に代入されると同時に、ステップS23で規定値 $h$ が加算され、タイミングT11でステップS24においてD/A変換器3に入力される。VCO2固有の発振最低周波数は、ロックレンジから充分離れているのでステップS25で、アンロック検出手段はアンロック信号を出力する。ステップS25、ステップS23、ステップS24においては、アンロック信号が出力されている間は、データ変更手段によってデータは規定値 $h$ ずつ増加させられ、周波数がロックされるまでこの動作を繰り返す。

【0040】ここで、規定値 $h$ は予想されるロックレンジの幅 $RL$ より少し小さい値に設定してある。

【0041】タイミングT12で、最初にロックしたデータS(N11')をステップS26においてデータD(N11)とし、それにステップS27で前記の規定値 $h$ より小さい規定値 $d$ を加算し、ステップS28でD/A変換器3に入力する。その後ステップS29、ステップS27、ステップS28で、規定値 $d$ の間隔でデータを加算しながらアンロック信号を監視する。そして、アンロック信号が出力されたタイミングT13の時点で、データ出力手段が出したデータD(N11')を、ステップS30でロックレンジの上端周波数データH(N11)として、RAM7に一時記憶させる。

【0042】続いて、ステップS31によって前記のデータD(N11)から規定値 $d$ を減算した後、タイミングT14でステップS32によってD/A変換器3に入力した後、ステップS32、ステップS33、ステップS31で、データを規定値 $d$ ずつ減算していった、アンロック信号の出力の様子を見る。アンロック信号が出力されたタイミングT15の時点で、データ出力手段が出したデータD(N11'')を、ステップS34でロックレンジの下端周波数データL(N11)として、RAM7に一時記憶させる。

【0043】その後、ステップS35でデータの平均化

の処理をし、ステップS36の書き込みの処理をする。プリセットすべき周波数が他にあれば、ステップS37によってステップS21にもどり、上述の動作を繰り返す。

【0044】本実施例のプリセット型PLL回路においては、データの変化量を初めは大きく、ロックレンジをみつければ小さく設定したので、早くロックレンジを見つけることができ、更に上端周波数と下端周波数が詳細な数値まで分かって、プリセットデータがより正確になる。

【0045】

【発明の効果】本発明のPLLプリセット方法およびプリセット型PLL回路によると、従来は必要であったVCOの入力電圧測定のためのA/D変換器の回路が不要になるので、配線および実装部品が少なくなって、低コスト化する上に、ハイインピーダンスであるVCOの入力から配線を引き出さなくてもよいので、PLL回路全体のC/N比が改善される。

【0046】また、プリセット型PLL回路を製作するに当たって、VCOやその他の構成素子にばらつきがあっても、所望の周波数のロックレンジの両端を正確に検出して得たデータを平均してプリセットデータとして、NデータとともにRAMに書き込むので、プリセットすべき周波数をより正確に発振させることができ、ロックアップタイムが短いプリセット型PLL回路を製作できる。更に量産した場合も不良品を最小限に抑えることができ、性能も向上する。

【図面の簡単な説明】

【図1】実施例1および実施例2のプリセット型PLL回路のブロック図である。

【図2】実施例1のプリセット型PLL回路のROMに内蔵された制御プログラムのフローチャート図である。

【図3】実施例1のプリセット型PLL回路のCPUの出力データとアンロック信号のタイムチャート図である。

【図4】実施例2のプリセット型PLL回路のROMに内蔵された制御プログラムのフローチャートであってその前半部を表す図である。

【図5】実施例2のプリセット型PLL回路のROMに内蔵された制御プログラムのフローチャートであって図4の前半部につながる後半部を表す図である。

【図6】実施例2のプリセット型PLL回路のCPUの出力データとアンロック信号のタイムチャート図である。

【図7】従来のプリセット型PLL回路の1例のブロック図である。

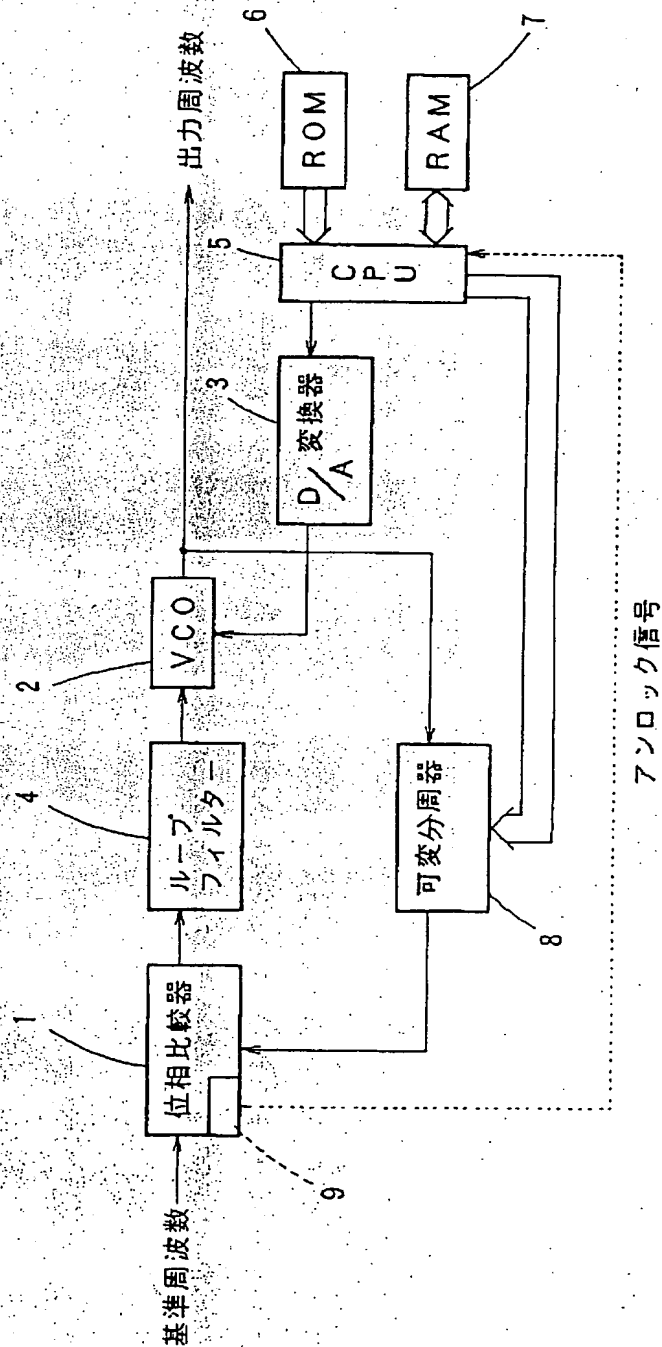
【符号の説明】

- 1 位相比較器
- 2 電圧制御型発振器(VCO)
- 3 デジタルアナログ変換器(D/A変換器)

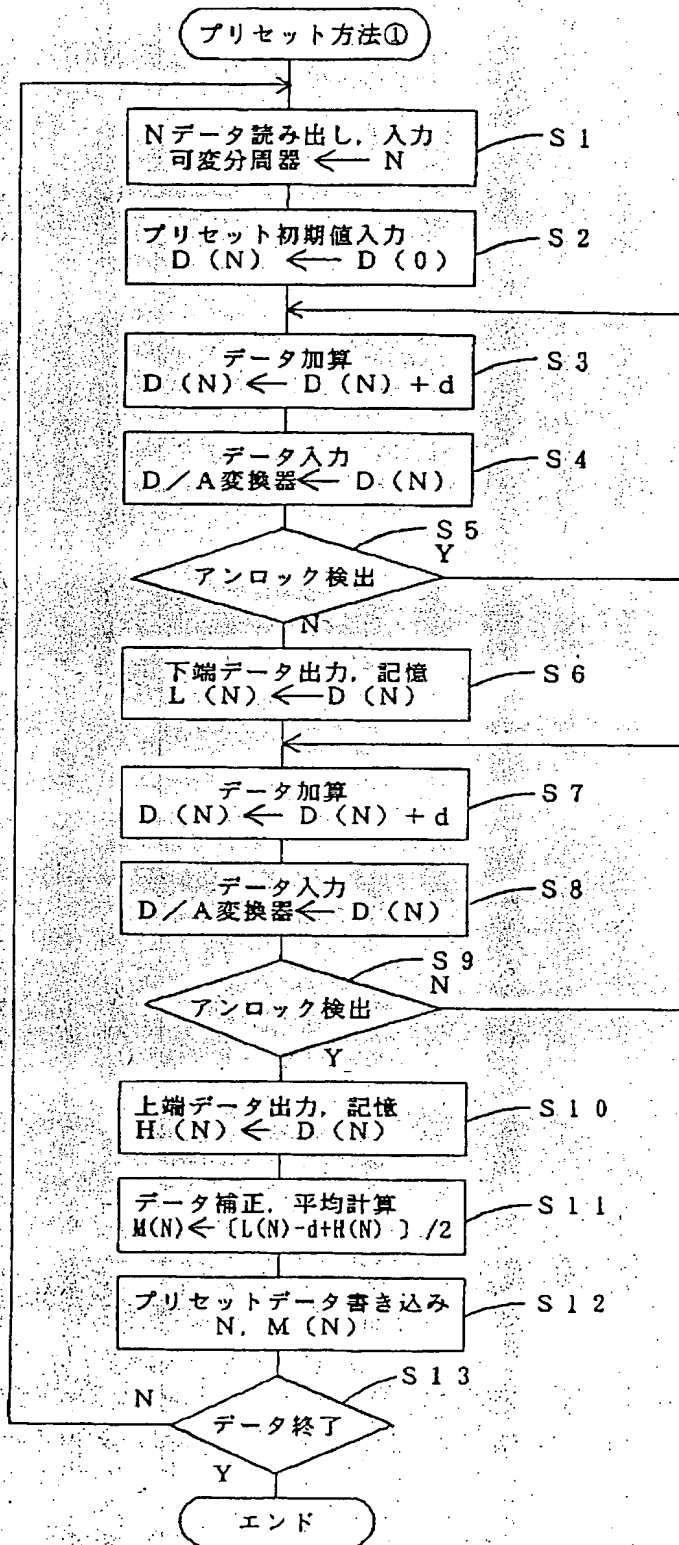
- 9  
4 ループフィルター  
5 マイクロプロセッサ (CPU)  
6 リードオンリーメモリー (ROM)  
7 ランダムアクセスメモリー (RAM) [記憶手段]  
8 可変分周器

- 10  
9 アンロック検出手段  
S3, S7, S23, S27, S31 データ変更手段  
S6, S10, S26, S30, S34 データ出力手段  
S11, S35 データ補正手段

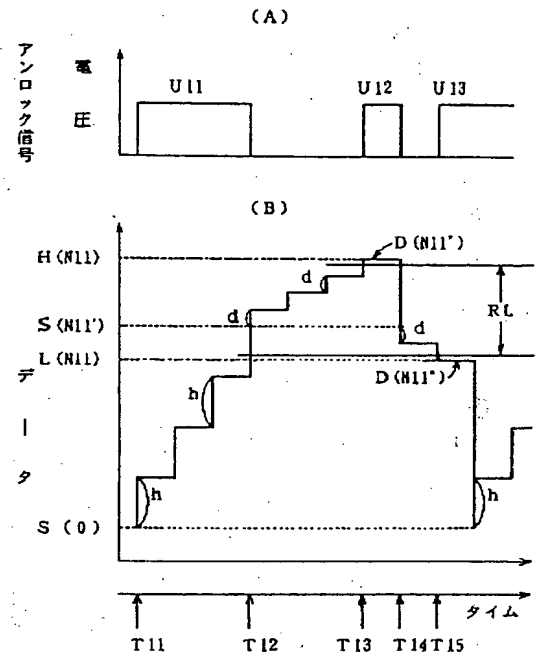
【図1】



【図2】

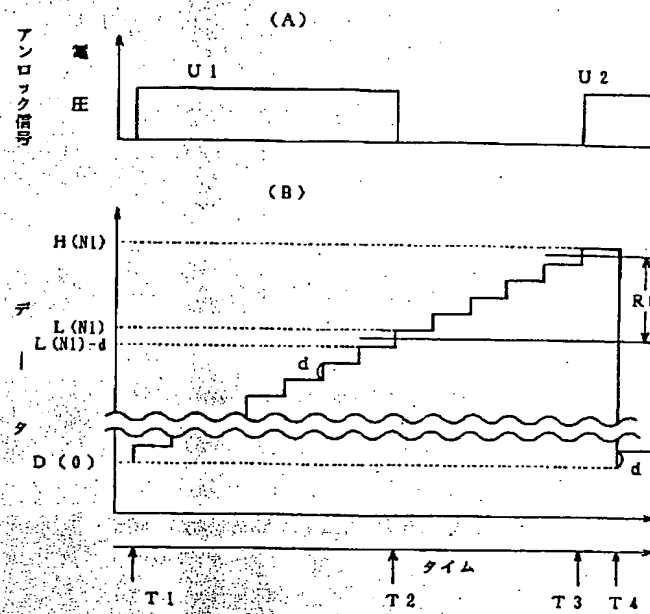


【図6】

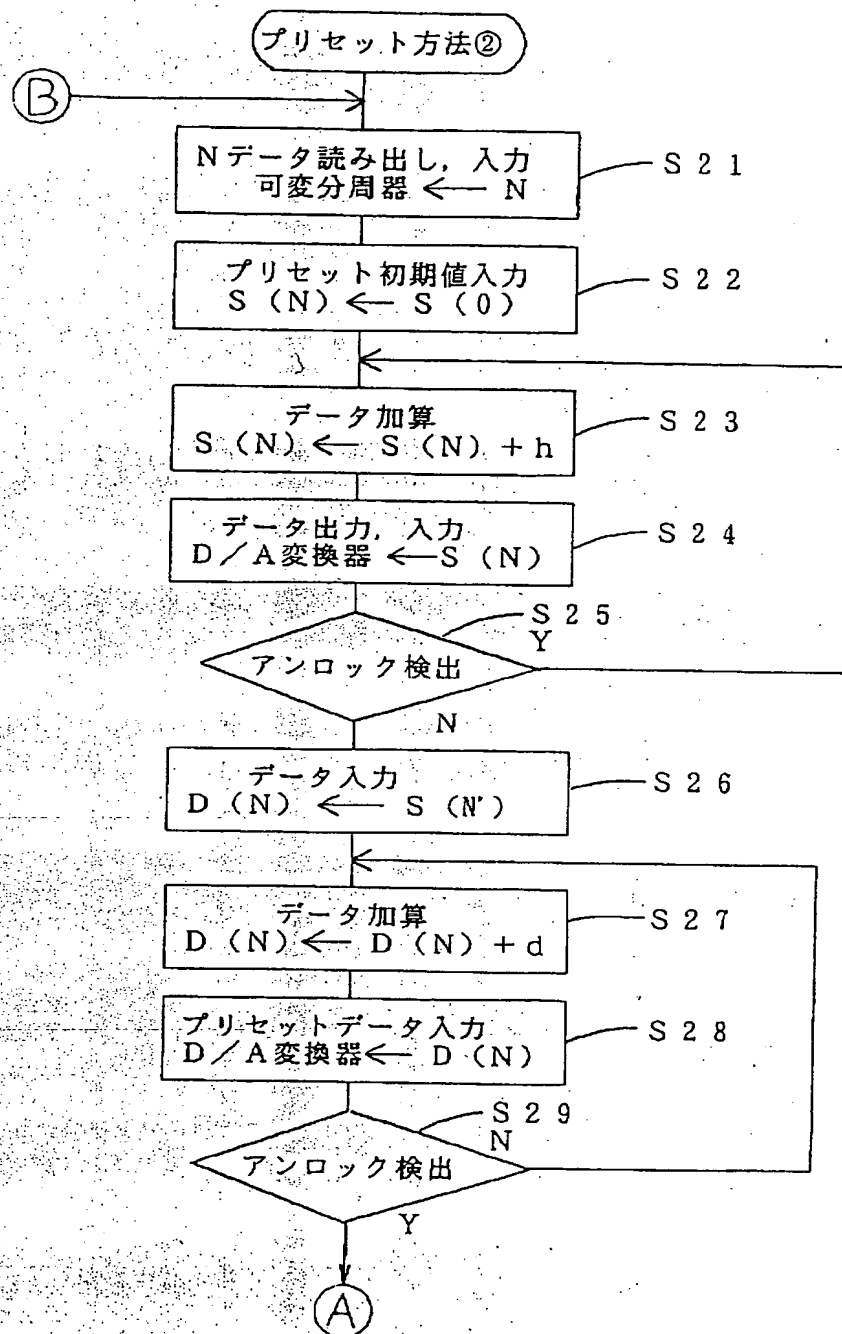




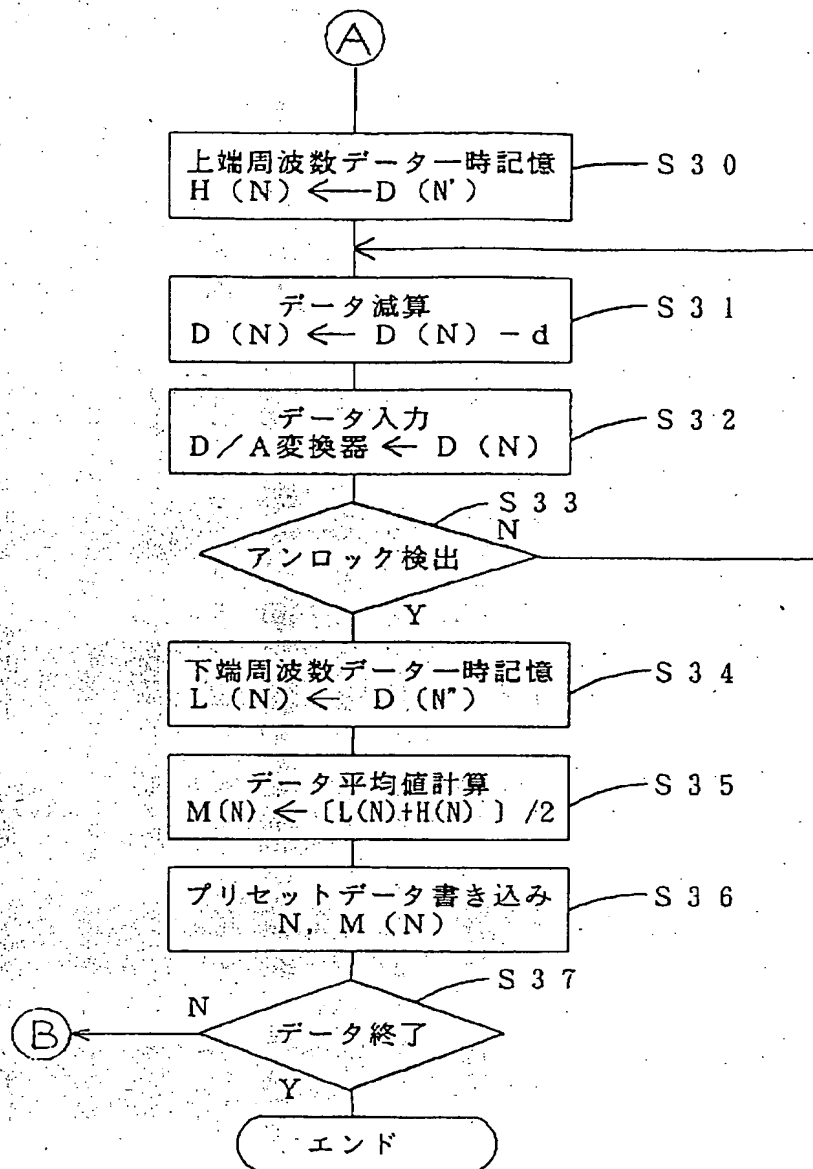
【図3】



【図4】



【図5】



【図7】

